

Horizontal Fusion Messaging Specification



**Department of Defense Assistant Secretary of Defense for
Networks and Information Integration/DOD CIO**

April 26, 2004



Revision Sheet

Release No.	Date	Revision Description
Ver. 1.2	4/14/2004	Messaging Specification



1	INTRODUCTION.....	1
1.1	Overview	1
1.1.1	Horizontal Fusion.....	1
1.1.2	Horizontal Fusion Portfolio Initiative.....	1
1.1.3	Quantum Leap Demonstrations	2
1.2	Purpose.....	2
2	MEDIATION/MESSAGING CES	2
2.1	Overview	2
2.2	WS-Eventing Specification.....	3
2.2.1	WS-Eventing Operations	3
2.2.2	Channel WSDL Definition	3
2.2.3	Publish Endpoint	3
2.2.4	Subscriber Endpoint.....	4
2.3	JMS Specification	5
2.4	Administration Specification	5
2.4.1	Bind.....	5
2.4.2	Rebind.....	5
2.4.3	Unbind.....	5
2.4.4	Search.....	6
2.4.5	SearchNext.....	6
2.4.6	Lookup	6
2.4.7	CreateTemp.....	6
2.5	Future	6
3	APPENDIX.....	6
3.1	WS-Eventing WSDL	6
3.2	Administration WSDL	8
3.2.1	Administration Types XSD.....	14



1 INTRODUCTION

1.1 Overview

1.1.1 Horizontal Fusion

Horizontal Fusion is a Net-Centric capability that enables ad hoc access by all echelons of users to data that is cross-functional and provides for decision superiority in the transformed battlespace.

Horizontal Fusion is a direct response to operationalize, integrate, and optimize technology and operations to achieve “Power to the Edge” in the new battlespace. Horizontal Fusion is made possible by the new technology context that includes Wideband Satellite Communications (SATCOM), the Global Information Grid Bandwidth Expansion (GIG-BE), and the Joint Tactical Radio System (JTRS). The increased bandwidth these capabilities will provide will deliver significant improvement in battlefield operations including real-time access to critical knowledge and enhanced situational awareness, informed and expeditious command decision-making, and operational and tactical response.

Moving beyond former concepts of “plug-and-play” wiring diagrams, “one-to-one” interfaces, or “push” technology, Horizontal Fusion is built on a foundation of transforming the ways in which knowledge is available in the battlespace.

The exploration of Net-Centric capabilities and the process necessary to provide them is the substance of Horizontal Fusion.

1.1.2 Horizontal Fusion Portfolio Initiative

The Horizontal Fusion Portfolio Initiative is a management process with an emphasis on outcome-based performance intended to accelerate the transition of Net-Centric warfighting capabilities into the operational inventory.

The Horizontal Fusion Portfolio is a set of technologies and processes that have been selected to demonstrate their interoperability with other members of the Portfolio and their contribution to the overall concept of Net-Centricity.

The intent is to accelerate the transition of Net-Centric warfighting from vision to reality. The Portfolio Initiative emphasizes outcome-based performance. Goals are closely aligned with the Department of Defense (DoD) vision of Transformation to Net-Centricity, with the objective of



building practical solutions. The Portfolio composition is guided by the overarching criteria of Fit, Balance, and Impact.

1.1.3 Quantum Leap Demonstrations

In August of each Fiscal Year, the Horizontal Fusion Portfolio conducts a demonstration of Net-Centricity, Interoperability, and Transformation capabilities that have been developed. Each year builds on the previous years' capabilities, continuously expanding the technology capability and increasing the decision superiority that is available to the Edge User.

1.2 Purpose

The purpose of this document is to document the Messaging Core Enterprise Service (CES)

2 MEDIATION/MESSAGING CES

2.1 Overview

The Mediation/Messaging CES provides a federated, distributed, and fault-tolerant enterprise message bus which:

- Delivers high performance, scalable and interoperable asynchronous event notification including alerts, track updates, etc. to both applications and end-users using multiple messaging models including publish and subscribe, queuing, and peer-to-peer
- Provides Quality of Service (QoS) including priority, precedence, and time-to-live
- Provides guaranteed delivery to disconnected users or applications
- Utilizes multiple message brokers, potentially within different administrative domains, to support the distributed, federated nature of the GIG

A number of standards have emerged which allow for the synthesis of a robust messaging system using web services. The foundations of the messaging specification rest with the Web Service Reliable Messaging [ws-rm] and Web Service Addressing [ws-addressing] standards that allow the expression of required QoS parameters for guaranteed delivery and decouple the destination of a message from the underlying transport, respectively. Both the WS-ReliableMessaging and WS-Addressing standards require support from the client web-service libraries. The WS-ReliableMessaging requires the client engage actively with the server whereas the WS-Addressing standard simply requires that the client set elements in the SOAP header. Client web-service libraries may automatically place the appropriate WS-Addressing elements in the header (for example: <wsa:Action/> and <wsa:MessageID/>). In lieu of client library support, the client application can simply add the header element where appropriate.



Built on these standards, two approaches to messaging have recently been put forth: Web Service Eventing [ws-eventing] and Web Service Notification [ws-notification]. WS-Eventing is a minimalist interface that was designed for adding notification to an existing web service. The ws-eventing standard only specifies operations for subscription since the specification assumes the end-point is the source of the events. An approach to developing a messaging broker that receives messages for delivery to subscribers needs to be added. The specification has a key advantage in that the publish and subscribe mechanism support arbitrary SOAP messages defined by administratively configured WSDLs. The WS-Notification is a more comprehensive specification. WS-Notification defines a comprehensive publish/subscribe architecture including concepts such as a message broker and potentially broker federation. The WS-Notification allows arbitrary messages to be published, but specifies specific operation to publish a message, effectively requiring applications to “wrap” the messages to conform to the WS-Notification WSDL. Unfortunately, since two consortiums submitted these specifications simultaneously, a consensus specification may emerge. For the near term, the simplest specification, WS-Eventing, was chosen as the basis for the messaging specification.

2.2 WS-Eventing Specification

2.2.1 WS-Eventing Operations

2.2.1.1 *Subscribe*

The subscribe web service establishes a durable subscription in the messaging service. The subscription will exist until either the user specified expiration occurs or the user un-subscribes from that subscription.

2.2.1.2 *Unsubscribe*

Un-subscribing from a subscription will cause the user to stop receiving messages across that topic.

2.2.1.3 *Renew*

Renewing a subscription prevents it from expiring. The renew web service allows the user to specify the expiration in either relative or absolute time. If a client fails to renew a subscription, then the client will be unsubscribed when the subscription expires. This prevents the consumption of resources by clients that are no longer active.

2.2.2 Channel WSDL Definition

The messaging service is able to accept and deliver any SOAP message as long as it contains the necessary delivery information in the SOAP header. It is assumed that the creator of a channel will assign a specific WSDL defining the content of messages on that channel.

2.2.3 Publish Endpoint

All publishing of messages to the messaging service is done via a single endpoint. This endpoint is called “Post” and knows which channel (topic) to use because of the necessary delivery



message in the SOAP header. The publisher uses the WSDL defined for the channel to generate an appropriately formatted message. The Message Service accepts arbitrary SOAP messages with properly formatted headers. The Messaging Service optionally can verify the message against the channel WSDL.

2.2.4 Subscriber Endpoint

Receiving of messages is done through two mechanisms, asynchronous subscription and synchronous subscription.

2.2.4.1 Asynchronous Subscription

Fundamentally, the WS-Eventing specification supports the asynchronous delivery model. When an event sink subscribes to a destination (the `<wsa:To/>` element) it specifies an end-point for the event broker to deliver events (specified by the `<wsa:NotifyTo/>` element). The returned information from a successful subscription provides a unique identifier for this subscription and a time when the subscription expires unless renewed. The contents of the `<wsa:ReferenceProperties/>` tag is copied from the subscription message allowing the event sink to supply its own identifier, or any other properties the event sink desires to be associated with this subscription.

At a later point in time an asynchronous event will be generated. Note that the `<wsa:ReferenceProperties/>` are again copied into the header. This asynchronous event will be delivered to the endpoint specified in the `<wsa:NotifyTo/>` element.

2.2.4.2 Synchronous Subscription

The asynchronous delivery case is the recommended approach for utilizing this API. However, asynchronous delivery requires a more complex consumer application. The event consumer must stand up a web-service end point capable of receiving the events. This requires the instantiation of a http server to process the asynchronous delivery. This changes the security posture of the sink significantly and requires additional security processing (since the consumer is now a “server”). Finally, if the consumer is behind a firewall, the operation may not be possible. For these reasons, a synchronous receive operation is supported.

The most appropriate approach to synchronous receive is currently being worked out with one of the web-service vendors, but a likely approach is that the WSDL for the synchronous receive is derived from the WSDL associated with the destination by transforming the “input” messages to “output” messages on all operations. The consumer then performs a request to a specified end-point on the broker and receives the events when they arrive. The consumer must still subscribe, but the “`<wsa:NotifyTo/>`” is not be supplied. The event consumer is still responsible for invoking the renew operation on a periodic basis. Finally, this approach requires the addition of a “timeout” either on the subscription operation or on each individual operation.



2.3 JMS Specification

An alternative to the WS-Eventing specification is the use of the Java Message Service (JMS) version 1.1. JMS is a well-known messaging standard, and is supported by multiple vendors. However, JMS is not sufficient for all messaging needs. The Mediation/Messaging CES supports the use of JMS as an interface to its messaging bus.

2.4 Administration Specification

With the adoption of WS-Eventing, the WSDL defining the messaging specification has been reduced to a specification of the administrative API. The administrative API allows clients to create entries such as topics, queues, and containers in a hierarchical namespace. An access control list is associated with the entries to control access. The access control lists may specify access based on the identity, group, or role to which the user belongs. Associated with a destination (a topic or queue) is an optional WSDL that specifies the message formats publishable to that destination. Other arbitrary metadata may be stored depending on the capability of the underlying provider.

There is an outstanding request for a notification API that would allow clients to register for notification of changes to the directory. Since this is the purpose of the WS-Eventing spec, a set of appropriate destinations will be configured to allow subscription for changes to the tree. The exact topic names are defined by the various COIs using the messaging service. The messaging service does not impose an explicit taxonomy for the destination names.

2.4.1 Bind

The bind operation creates a new entry in the directory. The success of the bind operation is contingent on having WRITE permissions to the parent of the entry being created and EXECUTE/SEARCH permissions to the root of the tree. The entry specifies permissions that may or may not override the permissions of the parent directories. The current “types” which are supported are “dir”, “topic” and “queue.” The WSDL specified will be utilized by the messaging system to enforce message formats. The operations within the WSDL must not have output or fault elements. Additional arbitrary metadata may be specified, and the system should store it or return a fault if unable (due to size limitations, etc).

2.4.2 Rebind

The re-bind operation either creates or replaces an entry in the directory. Otherwise, the semantics are the same as the bind operation.

2.4.3 Unbind

Removes the entry from the directory. The client must have WRITE permissions to the parent directory and EXECUTE/SEARCH permissions to the root of the tree.



2.4.4 Search

Search from a specified base entry over a defined scope (OBJECT, ONE_LEVEL, SUBTREE) returning all objects that match a specified filter. The filter is optional. If it is not present then all entries match. The client requires EXECUTE/SEARCH permissions to traverse the tree and READ permissions on an entry to return the entry. The entries must be returned in breadth-first order.

2.4.5 SearchNext

The search operation returns an initial set of entries and an identifier. To obtain more entries the searchNext operation is invoked with the identifier. The specified entries will be returned. The server may expire its cache of iterators, so the identifier will not be good indefinitely.

2.4.6 Lookup

Return the entry specified by the name. The client must have EXECUTE/SEARCH permissions from the root to the parent directory and READ permissions on the entry.

2.4.7 CreateTemp

The createTemp operation allows the creation of a destination that is private to a particular client (typically an event source). The destination is subscribed to by the client and used in the <wsa:ReplyTo/> header element of a published event to solicit a response from the recipients of the event. The event recipient publishes an event in response with the <wsa:To/> element set to the <wsa:ReplyTo/> element of the originating message. The <wsa:RefersTo/> element is set to the <wsa:MessageID/> element. The broker then will deliver the response back to the temporary destination consumed by the original subscriber. The temporary destination will be destroyed when the client fails to renew.

2.5 Future

A set of topics needs to be defined for notifying interested parties of changes within the administrative directory. Also, security requirements that arise because of the point-to-multi-point nature of the Messaging CES are still being researched.

3 APPENDIX

3.1 WS-Eventing WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wse="http://schemas.xmlsoap.org/ws/2004/01/eventing"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="urn:nces:messaging:1.2:service:eventing"
    targetNamespace="urn:nces:messaging:1.2:service:eventing">
    <wsdl:types>
        <xsd:schema>
```



```
<xs:import
namespace="http://schemas.xmlsoap.org/ws/2004/01/eventing"
schemaLocation="eventing.xsd"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="SubscribeMsg">
    <wsdl:part name="body" element="wse:Subscribe"/>
</wsdl:message>
<wsdl:message name="SubscribeResponseMsg">
    <wsdl:part name="body" element="wse:SubscribeResponse"/>
</wsdl:message>
<wsdl:message name="RenewMsg">
    <wsdl:part name="body" element="wse:Renew"/>
</wsdl:message>
<wsdl:message name="RenewResponseMsg">
    <wsdl:part name="body" element="wse:RenewResponse"/>
</wsdl:message>
<wsdl:message name="UnsubscribeMsg">
    <wsdl:part name="body" element="wse:Unsubscribe"/>
</wsdl:message>
<wsdl:message name="UnsubscribeResponseMsg"/>
<wsdl:portType name="Eventing">
    <wsdl:operation name="SubscribeOp">
        <wsdl:input message="tns:SubscribeMsg"/>
        <wsdl:output message="tns:SubscribeResponseMsg"/>
    </wsdl:operation>
    <!-- Support for Renew is optional -->
    <wsdl:operation name="RenewOp">
        <wsdl:input message="tns:RenewMsg"/>
        <wsdl:output message="tns:RenewResponseMsg"/>
    </wsdl:operation>
    <wsdl:operation name="UnsubscribeOp">
        <wsdl:input message="tns:UnsubscribeMsg"/>
        <wsdl:output message="tns:UnsubscribeResponseMsg"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="EventBinding" type="tns:Eventing">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="SubscribeOp">
        <soap:operation soapAction="urn:#SubscribeOp"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RenewOp">
        <soap:operation soapAction="urn:#RenewOp"
style="document"/>
        <wsdl:input>
```



```

        <soap:body use="literal"/>
    </wsdl:input>

    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="UnsubscribeOp">
    <soap:operation soapAction="urn:#UnsubscribeOp"
style="document"/>

    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Event">
    <wsdl:port name="Eventing" binding="tns:EventBinding">
        <soap:address location="http://localhost:6061"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

3.2 Administration WSDL

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="urn:nces:messaging:1.2:service:admin"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:ns="http://www.w3.org/2001/XMLSchema"
    xmlns:ns1="http://schemas.xmlsoap.org/ws/2003/03/addressing"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tns="urn:nces:messaging:1.2:service:admin">
    <types>
        <xss:schema attributeFormDefault="unqualified" blockDefault="#all"
            elementFormDefault="qualified"
            targetNamespace="urn:nces:messaging:1.2:service:admin"
version="0.1"
            xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
            xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
            xmlns:ns="http://www.w3.org/2001/XMLSchema"
            xmlns:admin="urn:nces:messaging:1.2:types:admin"
            xmlns:ns1="http://schemas.xmlsoap.org/ws/2003/03/addressing"
            xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
```



```
xmlns:tns="urn:nces:messaging:1.2:service:admin"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:import
    namespace="http://schemas.xmlsoap.org/ws/2003/03/addressing"
    schemaLocation="wsa.xsd"/>
<xs:import
    namespace="urn:nces:messaging:1.2:types:admin"
    schemaLocation="admin_types.xsd"/>
<xs:complexType name="SearchRequestType">
    <xs:sequence>
        <xs:element ref="admin:entryName"/>
        <xs:element minOccurs="0" ref="admin:scope"/>
        <xs:element minOccurs="0" name="filter"
type="xs:string"/>
        <xs:element minOccurs="0" name="maxResults"
type="xs:int"/>
        <xs:element minOccurs="0" name="timeout" type="xs:long"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="searchRequest" type="tns:SearchRequestType"/>
<xs:complexType name="SearchResponseType">
    <xs:sequence>
        <xs:element name="id" type="xs:string"/>
        <xs:element name="index" type="xs:positiveInteger"/>
        <xs:element name="maxIndex" type="xs:positiveInteger"/>
        <xs:element name="expiration" type="xs:dateTime"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
ref="admin:entry"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="searchResponse" type="tns:SearchResponseType"/>
<xs:complexType name="SearchNextRequestType">
    <xs:sequence>
        <xs:element name="id" type="xs:string"/>
        <xs:element name="index" type="xs:positiveInteger"/>
        <xs:element minOccurs="0" name="maxEntries"
type="xs:positiveInteger"/>
        <xs:element minOccurs="0" name="timeout" type="xs:long"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="searchNextRequest"
type="tns:SearchNextRequestType"/>
<xs:complexType name="CreateTempRequestType"/>
<xs:element name="createTempRequest"
type="tns:CreateTempRequestType"/>
<xs:complexType name="CreateTempResponseType">
    <xs:sequence>
        <xs:element ref="wsa:To"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="createTempResponse"
type="tns:CreateTempResponseType"/>
```



```
<xs:complexType name="BindRequestType">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="admin:entry"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="bindRequest" type="tns:BindRequestType"/>
<xs:complexType name="UnbindRequestType">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="admin:entryName"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="unbindRequest" type="tns:UnbindRequestType"/>
<xs:complexType name="RenameRequestType">
    <xs:sequence>
        <xs:element name="src" type="wsa:AttributedURI"/>
        <xs:element name="dest" type="wsa:AttributedURI"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="renameRequest" type="tns:RenameRequestType"/>
<xs:complexType name="LookupRequestType">
    <xs:sequence>
        <xs:element ref="admin:entryName"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="lookupRequest" type="tns:LookupRequestType"/>
<xs:complexType name="LookupResponseType">
    <xs:sequence>
        <xs:element ref="admin:entry"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="lookupResponse" type="tns:LookupResponseType"/>
<xs:complexType name="FaultType">
    <xs:sequence>
        <xs:element name="reason" type="xs:string"/>
        <xs:element minOccurs="0" name="errorcode"
type="xs:long"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="fault" type="tns:FaultType"/>
</xs:schema>
</types>
<message name="bindFault">
    <part element="tns:fault" name="fault"/>
</message>
<message name="bindInput">
    <part element="tns:bindRequest" name="bind"/>
</message>
<message name="createFault">
    <part element="tns:fault" name="fault"/>
</message>
<message name="createTempInput">
    <part element="tns:createTempRequest" name="createTemp"/>
</message>
```



```
<message name="createTempOutput">
    <part element="tns:createTempResponse" name="createTemp" />
</message>
<message name="empty" />
<message name="lookupFault">
    <part element="tns:fault" name="fault" />
</message>
<message name="lookupInput">
    <part element="tns:lookupRequest" name="lookup" />
</message>
<message name="lookupOutput">
    <part element="tns:lookupResponse" name="lookup" />
</message>
<message name="rebindFault">
    <part element="tns:fault" name="fault" />
</message>
<message name="rebindInput">
    <part element="tns:bindRequest" name="rebind" />
</message>
<message name="renameFault">
    <part element="tns:fault" name="fault" />
</message>
<message name="renameInput">
    <part element="tns:renameRequest" name="rename" />
</message>
<message name="searchFault">
    <part element="tns:fault" name="fault" />
</message>
<message name="searchInput">
    <part element="tns:searchRequest" name="search" />
</message>
<message name="searchNextFault">
    <part element="tns:fault" name="fault" />
</message>
<message name="searchNextInput">
    <part element="tns:searchNextRequest" name="searchNext" />
</message>
<message name="searchNextOutput">
    <part element="tns:searchResponse" name="searchNext" />
</message>
<message name="searchOutput">
    <part element="tns:searchResponse" name="search" />
</message>
<message name="unbindFault">
    <part element="tns:fault" name="fault" />
</message>
<message name="unbindInput">
    <part element="tns:unbindRequest" name="unbind" />
</message>
<portType name="AdminPortType">
    <operation name="bind">
        <input message="tns:bindInput" />
        <output message="tns:empty" />
```



```
<fault message="tns:bindFault" name="FaultName2" />
</operation>
<operation name="createTemp">
    <input message="tns:createTempInput"/>
    <output message="tns:createTempOutput"/>
    <fault message="tns:createFault" name="FaultName1" />
</operation>
<operation name="lookup">
    <input message="tns:lookupInput"/>
    <output message="tns:lookupOutput"/>
    <fault message="tns:lookupFault" name="FaultName6" />
</operation>
<operation name="rebind">
    <input message="tns:rebindInput"/>
    <output message="tns:empty"/>
    <fault message="tns:rebindFault" name="FaultName3" />
</operation>
<operation name="rename">
    <input message="tns:renameInput"/>
    <output message="tns:empty"/>
    <fault message="tns:renameFault" name="FaultName5" />
</operation>
<operation name="search">
    <input message="tns:searchInput"/>
    <output message="tns:searchOutput"/>
    <fault message="tns:searchFault" name="FaultName7" />
</operation>
<operation name="searchNext">
    <input message="tns:searchNextInput"/>
    <output message="tns:searchNextOutput"/>
    <fault message="tns:searchNextFault" name="FaultName8" />
</operation>
<operation name="unbind">
    <input message="tns:unbindInput"/>
    <output message="tns:empty"/>
    <fault message="tns:unbindFault" name="FaultName4" />
</operation>
</portType>
<binding name="AdminBinding" type="tns:AdminPortType">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="bind">
        <soap:operation soapAction="urn:#bind" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="FaultName2">
            <soap:fault name="FaultName2" use="literal" />
        </fault>
    </operation>
</binding>
```



```
<operation name="createTemp">
    <soap:operation soapAction="urn:#createTemp" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="FaultName1">
        <soap:fault name="FaultName1" use="literal" />
    </fault>
</operation>
<operation name="lookup">
    <soap:operation soapAction="urn:#lookup" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="FaultName6">
        <soap:fault name="FaultName6" use="literal" />
    </fault>
</operation>
<operation name="rebind">
    <soap:operation soapAction="urn:#rebind" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="FaultName3">
        <soap:fault name="FaultName3" use="literal" />
    </fault>
</operation>
<operation name="rename">
    <soap:operation soapAction="urn:#rename" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="FaultName5">
        <soap:fault name="FaultName5" use="literal" />
    </fault>
</operation>
<operation name="search">
    <soap:operation soapAction="urn:#search" />
    <input>
        <soap:body use="literal" />
    </input>
```



```
<output>
    <soap:body use="literal"/>
</output>
<fault name="FaultName7">
    <soap:fault name="FaultName7" use="literal"/>
</fault>
</operation>
<operation name="searchNext">
    <soap:operation soapAction="urn:#searchNext"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
    <fault name="FaultName8">
        <soap:fault name="FaultName8" use="literal"/>
    </fault>
</operation>
<operation name="unbind">
    <soap:operation soapAction="urn:#unbind"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
    <fault name="FaultName4">
        <soap:fault name="FaultName4" use="literal"/>
    </fault>
</operation>
</binding>
<service name="Admin">
    <port binding="tns:AdminBinding" name="AdminPort">
        <soap:address location="http://localhost:8080/admin"/>
    </port>
</service>
</definitions>
```

3.2.1 Administration Types XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by David LaVal
(Solers Inc) --&gt;
&lt;xss: schema targetNamespace="urn:nces:messaging:1.2:types:admin"
xmlns:admin="urn:nces:messaging:1.2:types:admin"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
elementFormDefault="unqualified" attributeFormDefault="unqualified"
version="2.0"&gt;
    &lt;xss:import namespace="http://schemas.xmlsoap.org/ws/2003/03/addressing"
schemaLocation="wsa.xsd"/&gt;</pre>
```



```
<xs:simpleType name="ScopeType">
    <xs:annotation>
        <xs:documentation>Scope for application of acl
entry</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="subtree"/>
        <xs:enumeration value="singlelevel"/>
        <xs:enumeration value="object"/>
    </xs:restriction>
</xs:simpleType>
<xs:element name="scope" type="admin:ScopeType" />
<xs:simpleType name="ACLEntryType" >
    <xs:annotation>
        <xs:documentation>Place holder for the allow/deny
entries</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string" />
</xs:simpleType>
<xs:complexType name="ACLTyype" >
    <xs:annotation>
        <xs:documentation>Specify the clients which can perform the
specified operation on this object</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" 
            name="member" type="admin:ACLEntryType" />
        <xs:element name="op" type="xs:string" />
        <xs:element name="scope" type="admin:ScopeType" />
        <xs:element default="true" name="allow" type="xs:boolean"
minOccurs="0" />
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="TypeType" >
    <xs:restriction base="xs:string">
        <xs:enumeration value="dir"/>
        <xs:enumeration value="entry"/>
    </xs:restriction>
</xs:simpleType>
<xs:element name="acl" type="admin:ACLTyype" />
<xs:element name="entryName" type="wsa:AttributedURI" />
<xs:complexType name="EntryType" >
    <xs:annotation>
        <xs:documentation>The entry being created, updated,
etc</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" ref="admin:acl" />
        <xs:element minOccurs="0" name="wsdl" type="wsa:AttributedURI" />
        <xs:element name="type" type="admin:TypeType" />
        <xs:element name="name" type="wsa:AttributedURI" />
        <xs:any namespace="##other" />
    </xs:sequence>
```



```
<xs:anyAttribute namespace="#other" />
</xs:complexType>
<xs:element name="entry" type="admin:EntryType" />
</xs:schema>
```



WSDL **eventing.wsdl**

WSDL location: <C:\Documents and Settings\skelley.SOLERS\Desktop\examples\wsdl\eventing.wsdl>
 targetnamespace: urn:nces:messaging:1.2:service:eventing

services	bindings	porttypes	messages
Event	EventBinding	Eventing	RenewMsg
			<u>RenewResponseMsg</u>
			<u>SubscribeMsg</u>
			<u>SubscribeResponseMsg</u>
			<u>UnsubscribeMsg</u>
			<u>UnsubscribeResponseMsg</u>

service **Event**

ports	Eventing binding <u>tns:EventBinding</u> extensibility <soap:address location="http://localhost:6061"/>
source	<wsdl:service name="Event"> <wsdl:port name="Eventing" binding="tns:EventBinding"> <soap:address location="http://localhost:6061"/> </wsdl:port> </wsdl:service>

binding **EventBinding**

type	<u>tns:Eventing</u>
extensibility	<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
operations	SubscribeOp extensibility <soap:operation soapAction="urn:#SubscribeOp"/> input <soap:body use="literal"/> output <soap:body use="literal"/> RenewOp extensibility <soap:operation soapAction="urn:#RenewOp" style="document"/> input <soap:body use="literal"/> output <soap:body use="literal"/> UnsubscribeOp extensibility <soap:operation soapAction="urn:#UnsubscribeOp" style="document"/> input <soap:body use="literal"/>



	output <soap:body use="literal"/>
used by	Service <u>Event</u> in Port <u>Eventing</u>
source	<pre> <wsdl:binding name="EventBinding" type="tns:Eventing"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="SubscribeOp"> <soap:operation soapAction="urn:#SubscribeOp"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="RenewOp"> <soap:operation soapAction="urn:#RenewOp" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="UnsubscribeOp"> <soap:operation soapAction="urn:#UnsubscribeOp" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> </pre>

porttype Eventing

operations	SubscribeOp <input type="button" value="input"/> <u>tns:SubscribeMsg</u> <input type="button" value="output"/> <u>tns:SubscribeResponseMsg</u> RenewOp <input type="button" value="input"/> <u>tns:RenewMsg</u> <input type="button" value="output"/> <u>tns:RenewResponseMsg</u> UnsubscribeOp <input type="button" value="input"/> <u>tns:UnsubscribeMsg</u> <input type="button" value="output"/> <u>tns:UnsubscribeResponseMsg</u>
used by	binding <u>EventBinding</u>
source	<pre> <wsdl:portType name="Eventing"> <wsdl:operation name="SubscribeOp"> <wsdl:input message="tns:SubscribeMsg"/> </wsdl:operation> </pre>



```
<wsdl:output message="tns:SubscribeResponseMsg"/>
</wsdl:operation>
<!-- Support for Renew is optional -->
<wsdl:operation name="RenewOp">
  <wsdl:input message="tns:RenewMsg"/>
  <wsdl:output message="tns:RenewResponseMsg"/>
</wsdl:operation>
<wsdl:operation name="UnsubscribeOp">
  <wsdl:input message="tns:UnsubscribeMsg"/>
  <wsdl:output message="tns:UnsubscribeResponseMsg"/>
</wsdl:operation>
</wsdl:portType>
```

message **SubscribeMsg**

parts	body element wse:Subscribe
used by	PortType Eventing in Operation SubscribeOp
source	<pre><wsdl:message name="SubscribeMsg"> <wsdl:part name="body" element="wse:Subscribe"/> </wsdl:message></pre>

message **SubscribeResponseMsg**

parts	body element wse:SubscribeResponse
used by	PortType Eventing in Operation SubscribeOp
source	<pre><wsdl:message name="SubscribeResponseMsg"> <wsdl:part name="body" element="wse:SubscribeResponse"/> </wsdl:message></pre>

message RenewMsg

parts	body element wse:Renew
used by	PortType Eventing in Operation RenewOp
source	<pre><wsdl:message name="RenewMsg"> <wsdl:part name="body" element="wse:Renew"/> </wsdl:message></pre>

message RenewResponseMsg

parts	body
-------	------



	element wse:RenewResponse
used by	PortType Eventing in Operation RenewOp
source	<wsdl:message name="RenewResponseMsg"> <wsdl:part name="body" element="wse:RenewResponse"/> </wsdl:message>

message **UnsubscribeMsg**

parts	body element wse:Unsubscribe
used by	PortType Eventing in Operation UnsubscribeOp
source	<wsdl:message name="UnsubscribeMsg"> <wsdl:part name="body" element="wse:Unsubscribe"/> </wsdl:message>

message **UnsubscribeResponseMsg**

parts	
used by	PortType Eventing in Operation UnsubscribeOp
source	<wsdl:message name="UnsubscribeResponseMsg"/>

WSDL documentation generated with **XMLSPY** WSDL Editor <http://www.altova.com/xmlspy>



Schema **eventing.wsdl**

WSDL location: <C:\Documents and Settings\skelley.SOLERS\Desktop\examples\wsdl\eventing.wsdl>
 targetnamespace: urn:nces:messaging:1.2:service:eventing

services	bindings	porttypes	messages
Event	EventBinding	Eventing	RenewMsg
			<u>RenewResponseMsg</u>
			<u>SubscribeMsg</u>
			<u>SubscribeResponseMsg</u>
			<u>UnsubscribeMsg</u>
			<u>UnsubscribeResponseMsg</u>

service **Event**

ports	Eventing binding <u>tns:EventBinding</u> extensibility <soap:address location="http://localhost:6061"/>
source	<wsdl:service name="Event"> <wsdl:port name="Eventing" binding="tns:EventBinding"> <soap:address location="http://localhost:6061"/> </wsdl:port> </wsdl:service>

binding **EventBinding**

type	<u>tns:Eventing</u>
extensibility	<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
operations	SubscribeOp extensibility <soap:operation soapAction="urn:#SubscribeOp"/> input <soap:body use="literal"/> output <soap:body use="literal"/> RenewOp extensibility <soap:operation soapAction="urn:#RenewOp" style="document"/> input <soap:body use="literal"/> output <soap:body use="literal"/> UnsubscribeOp extensibility <soap:operation soapAction="urn:#UnsubscribeOp" style="document"/> input <soap:body use="literal"/>



	output <soap:body use="literal"/>
used by	Service <u>Event</u> in Port <u>Eventing</u>
source	<pre> <wsdl:binding name="EventBinding" type="tns:Eventing"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="SubscribeOp"> <soap:operation soapAction="urn:#SubscribeOp"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="RenewOp"> <soap:operation soapAction="urn:#RenewOp" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="UnsubscribeOp"> <soap:operation soapAction="urn:#UnsubscribeOp" style="document"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> </pre>

porttype Eventing

operations	SubscribeOp <input type="button" value="input"/> <u>tns:SubscribeMsg</u> <input type="button" value="output"/> <u>tns:SubscribeResponseMsg</u> RenewOp <input type="button" value="input"/> <u>tns:RenewMsg</u> <input type="button" value="output"/> <u>tns:RenewResponseMsg</u> UnsubscribeOp <input type="button" value="input"/> <u>tns:UnsubscribeMsg</u> <input type="button" value="output"/> <u>tns:UnsubscribeResponseMsg</u>
used by	binding <u>EventBinding</u>
source	<pre> <wsdl:portType name="Eventing"> <wsdl:operation name="SubscribeOp"> <wsdl:input message="tns:SubscribeMsg"/> </wsdl:operation> <wsdl:operation name="RenewOp"> <wsdl:input message="tns:RenewMsg"/> </wsdl:operation> <wsdl:operation name="UnsubscribeOp"> <wsdl:input message="tns:UnsubscribeMsg"/> </wsdl:operation> </wsdl:portType> </pre>



	<pre> <wsdl:output message="tns:SubscribeResponseMsg"/> </wsdl:operation> <!-- Support for Renew is optional --> <wsdl:operation name="RenewOp"> <wsdl:input message="tns:RenewMsg"/> <wsdl:output message="tns:RenewResponseMsg"/> </wsdl:operation> <wsdl:operation name="UnsubscribeOp"> <wsdl:input message="tns:UnsubscribeMsg"/> <wsdl:output message="tns:UnsubscribeResponseMsg"/> </wsdl:operation> </wsdl:portType> </pre>
--	--

message **SubscribeMsg**

parts	body element wse:Subscribe
used by	PortType Eventing in Operation SubscribeOp
source	<pre><wsdl:message name="SubscribeMsg"> <wsdl:part name="body" element="wse:Subscribe"/> </wsdl:message></pre>

message **SubscribeResponseMsg**

parts	body element wse:SubscribeResponse
used by	PortType Eventing in Operation SubscribeOp
source	<pre><wsdl:message name="SubscribeResponseMsg"> <wsdl:part name="body" element="wse:SubscribeResponse"/> </wsdl:message></pre>

message **RenewMsg**

parts	body element wse:Renew
used by	PortType Eventing in Operation RenewOp
source	<pre><wsdl:message name="RenewMsg"> <wsdl:part name="body" element="wse:Renew"/> </wsdl:message></pre>

message **RenewResponseMsg**

parts	body



	element wse:RenewResponse
used by	PortType Eventing in Operation RenewOp
source	<wsdl:message name="RenewResponseMsg"> <wsdl:part name="body" element="wse:RenewResponse"/> </wsdl:message>

message **UnsubscribeMsg**

parts	body element wse:Unsubscribe
used by	PortType Eventing in Operation UnsubscribeOp
source	<wsdl:message name="UnsubscribeMsg"> <wsdl:part name="body" element="wse:Unsubscribe"/> </wsdl:message>

message **UnsubscribeResponseMsg**

parts	
used by	PortType Eventing in Operation UnsubscribeOp
source	<wsdl:message name="UnsubscribeResponseMsg"/>

WSDL documentation generated with **XMLSPY** WSDL Editor <http://www.altova.com/xmlspy>



WSDL admin.wsdl

WSDL location: <C:\Documents and Settings\skelley.SOLERS\Desktop\examples\wsdl\admin.wsdl>
 targetnamespace: urn:nces:messaging:1.2:service:admin

services	bindings	porttypes	messages
Admin			
Admin			
AdminBinding			
AdminPortType			
bindFault			
bindInput			
createFault			
createTempInput			
createTempOutput			
empty			
lookupFault			
lookupInput			
lookupOutput			
rebindFault			
rebindInput			
renameFault			
renameInput			
searchFault			
searchInput			
searchNextFault			
searchNextInput			
searchNextOutput			
searchOutput			
unbindFault			
unbindInput			

service Admin

ports	AdminPort <table border="0"> <tr> <td style="vertical-align: top;">binding</td><td>tns:AdminBinding</td></tr> <tr> <td style="vertical-align: top;">extensibility</td><td><soap:address location="http://localhost:8080/admin"/></td></tr> </table>	binding	tns:AdminBinding	extensibility	<soap:address location="http://localhost:8080/admin"/>
binding	tns:AdminBinding				
extensibility	<soap:address location="http://localhost:8080/admin"/>				
source	<service name="Admin"> <port name="AdminPort" binding="tns:AdminBinding"> <soap:address location="http://localhost:8080/admin"/> </port> </service>				

binding AdminBinding



type	<u>tns:AdminPortType</u>																																														
extensibility	<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>																																														
operations	<p>bind</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#bind"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> <tr> <td style="background-color: #e0e0e0;">output</td><td><soap:body use="literal"/></td></tr> </table> <p>createTemp</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#createTemp"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> <tr> <td style="background-color: #e0e0e0;">output</td><td><soap:body use="literal"/></td></tr> </table> <p>lookup</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#lookup"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> <tr> <td style="background-color: #e0e0e0;">output</td><td><soap:body use="literal"/></td></tr> </table> <p>rebind</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#rebind"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> <tr> <td style="background-color: #e0e0e0;">output</td><td><soap:body use="literal"/></td></tr> </table> <p>rename</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#rename"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> <tr> <td style="background-color: #e0e0e0;">output</td><td><soap:body use="literal"/></td></tr> </table> <p>search</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#search"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> <tr> <td style="background-color: #e0e0e0;">output</td><td><soap:body use="literal"/></td></tr> </table> <p>searchNext</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#searchNext"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> <tr> <td style="background-color: #e0e0e0;">output</td><td><soap:body use="literal"/></td></tr> </table> <p>unbind</p> <table> <tr> <td style="background-color: #e0e0e0;">extensibility</td><td><soap:operation soapAction="urn:#unbind"/></td></tr> <tr> <td style="background-color: #e0e0e0;">input</td><td><soap:body use="literal"/></td></tr> </table>	extensibility	<soap:operation soapAction="urn:#bind"/>	input	<soap:body use="literal"/>	output	<soap:body use="literal"/>	extensibility	<soap:operation soapAction="urn:#createTemp"/>	input	<soap:body use="literal"/>	output	<soap:body use="literal"/>	extensibility	<soap:operation soapAction="urn:#lookup"/>	input	<soap:body use="literal"/>	output	<soap:body use="literal"/>	extensibility	<soap:operation soapAction="urn:#rebind"/>	input	<soap:body use="literal"/>	output	<soap:body use="literal"/>	extensibility	<soap:operation soapAction="urn:#rename"/>	input	<soap:body use="literal"/>	output	<soap:body use="literal"/>	extensibility	<soap:operation soapAction="urn:#search"/>	input	<soap:body use="literal"/>	output	<soap:body use="literal"/>	extensibility	<soap:operation soapAction="urn:#searchNext"/>	input	<soap:body use="literal"/>	output	<soap:body use="literal"/>	extensibility	<soap:operation soapAction="urn:#unbind"/>	input	<soap:body use="literal"/>
extensibility	<soap:operation soapAction="urn:#bind"/>																																														
input	<soap:body use="literal"/>																																														
output	<soap:body use="literal"/>																																														
extensibility	<soap:operation soapAction="urn:#createTemp"/>																																														
input	<soap:body use="literal"/>																																														
output	<soap:body use="literal"/>																																														
extensibility	<soap:operation soapAction="urn:#lookup"/>																																														
input	<soap:body use="literal"/>																																														
output	<soap:body use="literal"/>																																														
extensibility	<soap:operation soapAction="urn:#rebind"/>																																														
input	<soap:body use="literal"/>																																														
output	<soap:body use="literal"/>																																														
extensibility	<soap:operation soapAction="urn:#rename"/>																																														
input	<soap:body use="literal"/>																																														
output	<soap:body use="literal"/>																																														
extensibility	<soap:operation soapAction="urn:#search"/>																																														
input	<soap:body use="literal"/>																																														
output	<soap:body use="literal"/>																																														
extensibility	<soap:operation soapAction="urn:#searchNext"/>																																														
input	<soap:body use="literal"/>																																														
output	<soap:body use="literal"/>																																														
extensibility	<soap:operation soapAction="urn:#unbind"/>																																														
input	<soap:body use="literal"/>																																														



	output <soap:body use="literal"/>
used by	Service <u>Admin</u> in Port <u>AdminPort</u>
source	<pre> <binding name="AdminBinding" type="tns:AdminPortType"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="bind"> <soap:operation soapAction="urn:#bind"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <fault name="FaultName2"> <soap:fault name="FaultName2" use="literal"/> </fault> </operation> <operation name="createTemp"> <soap:operation soapAction="urn:#createTemp"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <fault name="FaultName1"> <soap:fault name="FaultName1" use="literal"/> </fault> </operation> <operation name="lookup"> <soap:operation soapAction="urn:#lookup"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <fault name="FaultName6"> <soap:fault name="FaultName6" use="literal"/> </fault> </operation> <operation name="rebind"> <soap:operation soapAction="urn:#rebind"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <fault name="FaultName3"> <soap:fault name="FaultName3" use="literal"/> </fault> </operation> <operation name="rename"> <soap:operation soapAction="urn:#rename"/> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> <fault name="FaultName5"> </pre>



```

<soap:fault name="FaultName5" use="literal"/>
</fault>
</operation>
<operation name="search">
<soap:operation soapAction="urn:#search"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="FaultName7">
<soap:fault name="FaultName7" use="literal"/>
</fault>
</operation>
<operation name="searchNext">
<soap:operation soapAction="urn:#searchNext"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="FaultName8">
<soap:fault name="FaultName8" use="literal"/>
</fault>
</operation>
<operation name="unbind">
<soap:operation soapAction="urn:#unbind"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="FaultName4">
<soap:fault name="FaultName4" use="literal"/>
</fault>
</operation>
</binding>

```

porttype AdminPortType

operations	bind <input type="button" value="input"/> <u>tns:bindInput</u>
------------	--



	<table border="1"> <tr> <td>output</td><td><u>tns:lookupOutput</u></td></tr> <tr> <td>fault</td><td><u>tns:lookupFault</u></td></tr> </table>	output	<u>tns:lookupOutput</u>	fault	<u>tns:lookupFault</u>				
output	<u>tns:lookupOutput</u>								
fault	<u>tns:lookupFault</u>								
	<table border="1"> <tr> <td>rebind</td><td></td></tr> <tr> <td>input</td><td><u>tns:rebindInput</u></td></tr> <tr> <td>output</td><td><u>tns:empty</u></td></tr> <tr> <td>fault</td><td><u>tns:rebindFault</u></td></tr> </table>	rebind		input	<u>tns:rebindInput</u>	output	<u>tns:empty</u>	fault	<u>tns:rebindFault</u>
rebind									
input	<u>tns:rebindInput</u>								
output	<u>tns:empty</u>								
fault	<u>tns:rebindFault</u>								
	<table border="1"> <tr> <td>rename</td><td></td></tr> <tr> <td>input</td><td><u>tns:renameInput</u></td></tr> <tr> <td>output</td><td><u>tns:empty</u></td></tr> <tr> <td>fault</td><td><u>tns:renameFault</u></td></tr> </table>	rename		input	<u>tns:renameInput</u>	output	<u>tns:empty</u>	fault	<u>tns:renameFault</u>
rename									
input	<u>tns:renameInput</u>								
output	<u>tns:empty</u>								
fault	<u>tns:renameFault</u>								
	<table border="1"> <tr> <td>search</td><td></td></tr> <tr> <td>input</td><td><u>tns:searchInput</u></td></tr> <tr> <td>output</td><td><u>tns:searchOutput</u></td></tr> <tr> <td>fault</td><td><u>tns:searchFault</u></td></tr> </table>	search		input	<u>tns:searchInput</u>	output	<u>tns:searchOutput</u>	fault	<u>tns:searchFault</u>
search									
input	<u>tns:searchInput</u>								
output	<u>tns:searchOutput</u>								
fault	<u>tns:searchFault</u>								
	<table border="1"> <tr> <td>searchNext</td><td></td></tr> <tr> <td>input</td><td><u>tns:searchNextInput</u></td></tr> <tr> <td>output</td><td><u>tns:searchNextOutput</u></td></tr> <tr> <td>fault</td><td><u>tns:searchNextFault</u></td></tr> </table>	searchNext		input	<u>tns:searchNextInput</u>	output	<u>tns:searchNextOutput</u>	fault	<u>tns:searchNextFault</u>
searchNext									
input	<u>tns:searchNextInput</u>								
output	<u>tns:searchNextOutput</u>								
fault	<u>tns:searchNextFault</u>								
	<table border="1"> <tr> <td>unbind</td><td></td></tr> <tr> <td>input</td><td><u>tns:unbindInput</u></td></tr> <tr> <td>output</td><td><u>tns:empty</u></td></tr> <tr> <td>fault</td><td><u>tns:unbindFault</u></td></tr> </table>	unbind		input	<u>tns:unbindInput</u>	output	<u>tns:empty</u>	fault	<u>tns:unbindFault</u>
unbind									
input	<u>tns:unbindInput</u>								
output	<u>tns:empty</u>								
fault	<u>tns:unbindFault</u>								
used by	binding <u>AdminBinding</u>								
source	<pre><portType name="AdminPortType"> <operation name="bind"> <input message="tns:bindInput"/> <output message="tns:empty"/> <fault name="FaultName2" message="tns:bindFault"/> </operation> <operation name="createTemp"> <input message="tns:createTempInput"/> <output message="tns:createTempOutput"/> <fault name="FaultName1" message="tns:createFault"/> </operation> <operation name="lookup"> <input message="tns:lookupInput"/> <output message="tns:lookupOutput"/> <fault name="FaultName6" message="tns:lookupFault"/> </operation> <operation name="rebind"> <input message="tns:rebindInput"/></pre>								



```

<output message="tns:empty"/>
<fault name="FaultName3" message="tns:rebindFault"/>
</operation>
<operation name="rename">
<input message="tns:renameInput"/>
<output message="tns:empty"/>
<fault name="FaultName5" message="tns:renameFault"/>
</operation>
<operation name="search">
<input message="tns:searchInput"/>
<output message="tns:searchOutput"/>
<fault name="FaultName7" message="tns:searchFault"/>
</operation>
<operation name="searchNext">
<input message="tns:searchNextInput"/>
<output message="tns:searchNextOutput"/>
<fault name="FaultName8" message="tns:searchNextFault"/>
</operation>
<operation name="unbind">
<input message="tns:unbindInput"/>
<output message="tns:empty"/>
<fault name="FaultName4" message="tns:unbindFault"/>
</operation>
</portType>

```

message bindFault

parts	fault element tns:fault
used by	PortType AdminPortType in Operation bind
source	<message name="bindFault"> <part name="fault" element="tns:fault"/> </message>

message bindInput

parts	bind element tns:bindRequest
used by	PortType AdminPortType in Operation bind
source	<message name="bindInput"> <part name="bind" element="tns:bindRequest"/> </message>

message createFault

parts	fault element tns:fault
used by	PortType AdminPortType in Operation createTemp



source	<pre><message name="createFault"> <part name="fault" element="tns:fault"/> </message></pre>
--------	---

message **createTempInput**

parts	createTemp element tns:createTempRequest
used by	PortType AdminPortType in Operation createTemp
source	<pre><message name="createTempInput"> <part name="createTemp" element="tns:createTempRequest"/> </message></pre>

message **createTempOutput**

parts	createTemp element tns:createTempResponse
used by	PortType AdminPortType in Operation createTemp
source	<pre><message name="createTempOutput"> <part name="createTemp" element="tns:createTempResponse"/> </message></pre>

message **empty**

parts	
used by	PortType AdminPortType in Operation bind PortType AdminPortType in Operation rebind PortType AdminPortType in Operation rename PortType AdminPortType in Operation unbind
source	<pre><message name="empty"/></pre>

message **lookupFault**

parts	fault element tns:fault
used by	PortType AdminPortType in Operation lookup
source	<pre><message name="lookupFault"> <part name="fault" element="tns:fault"/> </message></pre>



message **lookupInput**

parts	lookup element tns:lookupRequest
used by	PortType AdminPortType in Operation lookup
source	<message name="lookupInput"> <part name="lookup" element="tns:lookupRequest"/> </message>

message **lookupOutput**

parts	lookup element tns:lookupResponse
used by	PortType AdminPortType in Operation lookup
source	<message name="lookupOutput"> <part name="lookup" element="tns:lookupResponse"/> </message>

message **rebindFault**

parts	fault element tns:fault
used by	PortType AdminPortType in Operation rebind
source	<message name="rebindFault"> <part name="fault" element="tns:fault"/> </message>

message **rebindInput**

parts	rebind element tns:bindRequest
used by	PortType AdminPortType in Operation rebind
source	<message name="rebindInput"> <part name="rebind" element="tns:bindRequest"/> </message>

message **renameFault**

parts	fault
-------	--------------



	element tns:fault
used by	PortType AdminPortType in Operation rename
source	<message name="renameFault"> <part name="fault" element="tns:fault"/> </message>

message renameInput

parts	rename element tns:renameRequest
used by	PortType AdminPortType in Operation rename
source	<message name="renameInput"> <part name="rename" element="tns:renameRequest"/> </message>

message searchFault

parts	fault element tns:fault
used by	PortType AdminPortType in Operation search
source	<message name="searchFault"> <part name="fault" element="tns:fault"/> </message>

message searchInput

parts	search element tns:searchRequest
used by	PortType AdminPortType in Operation search
source	<message name="searchInput"> <part name="search" element="tns:searchRequest"/> </message>

message searchNextFault

parts	fault element tns:fault
used by	PortType AdminPortType in Operation searchNext



source	<pre><message name="searchNextFault"> <part name="fault" element="tns:fault"/> </message></pre>
--------	---

message **searchNextInput**

parts	searchNext element tns:searchNextRequest
used by	PortType AdminPortType in Operation searchNext
source	<pre><message name="searchNextInput"> <part name="searchNext" element="tns:searchNextRequest"/> </message></pre>

message **searchNextOutput**

parts	searchNext element tns:searchResponse
used by	PortType AdminPortType in Operation searchNext
source	<pre><message name="searchNextOutput"> <part name="searchNext" element="tns:searchResponse"/> </message></pre>

message **searchOutput**

parts	search element tns:searchResponse
used by	PortType AdminPortType in Operation search
source	<pre><message name="searchOutput"> <part name="search" element="tns:searchResponse"/> </message></pre>

message **unbindFault**

parts	fault element tns:fault
used by	PortType AdminPortType in Operation unbind
source	<pre><message name="unbindFault"> <part name="fault" element="tns:fault"/> </message></pre>

**message unbindInput**

parts	unbind element tns:unbindRequest
used by	PortType AdminPortType in Operation unbind
source	<message name="unbindInput"> <part name="unbind" element="tns:unbindRequest"/> </message>

WSDL documentation generated with **XMLSPY** WSDL Editor <http://www.altova.com/xmlspy>



Schema `admin_types.xsd`

schema location: C:\Documents and Settings\skelley.SOLERS\Desktop\examples\wsdl\admin_types.xsd
 targetNamespace: `urn:nces:messaging:1.2:types:admin`

Elements Complex types Simple types

<code>acl</code>	ACLType	ACLEntryType
<code>entry</code>	EntryType	EntryNameType
<code>entryName</code>		ScopeType
<code>scope</code>		TypeType

schema location: <C:\Documents and Settings\skelley.SOLERS\Desktop\examples\wsdl\wsa.xsd>
 targetNamespace: <http://schemas.xmlsoap.org/ws/2003/03/addressing>

Elements Complex types Simple types

Action	AttributedQName	RelationshipTypeValues
EndpointReference	AttributedURI	
FaultTo		EndpointReferenceType
From		ReferencePropertiesType
MessageID		Relationship
Recipient		ServiceNameType
RelatesTo		
ReplyTo		
To		

element `acl`

diagram	<pre> classDiagram class acl class op class member class scope class allow acl "3..1" -- "0..1" op op "0..1" -- "0..1" member op "0..1" -- "0..1" scope scope "0..1" -- "0..1" allow </pre>
namespace	<code>urn:nces:messaging:1.2:types:admin</code>
type	admin:ACLType
children	member op scope allow



used by	complexType <u>EntryType</u>
source	<xs:element name="acl" type="admin:ACLType"/>

element entry

diagram	<pre> classDiagram class admin:EntryType { entry +acl wsdl type name any #other } entry <--> admin:EntryType acl <--> admin:EntryType wsdl <--> admin:EntryType type <--> admin:EntryType name <--> admin:EntryType any #other <--> admin:EntryType </pre>
namespace	urn:nces:messaging:1.2:types:admin
type	<u>admin:EntryType</u>
children	<u>admin:acl</u> <u>wsdl</u> <u>type</u> <u>name</u>
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="entry" type="admin:EntryType"/>

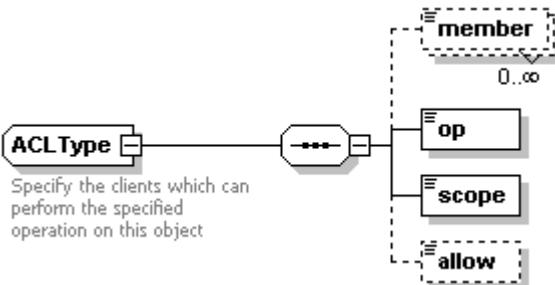
element entryName

diagram	<pre> classDiagram class entryName </pre>
namespace	urn:nces:messaging:1.2:types:admin
type	<u>admin:EntryNameType</u>
source	<xs:element name="entryName" type="admin:EntryNameType"/>

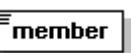
element scope

diagram	<pre> classDiagram class scope </pre>
namespace	urn:nces:messaging:1.2:types:admin
type	<u>admin:ScopeType</u>
facets	enumeration subtree enumeration singlelevel enumeration object
source	<xs:element name="scope" type="admin:ScopeType"/>

complexType ACLType

diagram	 <p>ACLType</p> <p>Specify the clients which can perform the specified operation on this object</p>
namespace	urn:nces:messaging:1.2:types:admin
children	member op scope allow
used by	element acl
annotation	documentation Specify the clients which can perform the specified operation on this object
source	<pre><xs:complexType name="ACLType"> <xs:annotation> <xs:documentation>Specify the clients which can perform the specified operation on this object</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="member" type="admin:ACLEntryType" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="op" type="xs:string"/> <xs:element name="scope" type="admin:ScopeType"/> <xs:element name="allow" type="xs:boolean" default="true" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

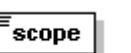
element ACLType/member

diagram	
type	admin:ACLEntryType
source	<pre><xs:element name="member" type="admin:ACLEntryType" minOccurs="0" maxOccurs="unbounded"/></pre>

element ACLType/op

diagram	
type	xs:string
source	<pre><xs:element name="op" type="xs:string"/></pre>

element ACLType/scope

diagram	
---------	---



type	admin:ScopeType
facets	enumeration subtree enumeration singlelevel enumeration object
source	<xs:element name="scope" type="admin:ScopeType"/>

element **ACLTType/allow**

diagram	
type	xs:boolean
source	<xs:element name="allow" type="xs:boolean" default="true" minOccurs="0"/>

complexType **EntryType**

diagram	
namespace	urn:nces:messaging:1.2:types:admin
children	admin:acl wsdl type name
used by	element entry
attributes	Name Type Use Default Fixed Annotation
annotation	documentation: The entry being created, updated, etc
source	<pre> <xs:complexType name="EntryType"> <xs:annotation> <xs:documentation>The entry being created, updated, etc</xs:documentation> </xs:annotation> <xs:sequence> <xs:element ref="admin:acl" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="wsdl" type="wsa:AttributedURI" minOccurs="0"/> <xs:element name="type" type="admin:TypeType"/> <xs:element name="name" type="admin:EntryNameType"/> <xs:any namespace="##other"/> </xs:sequence> </xs:complexType> </pre>

element **EntryType/wsdl**



diagram	
type	<u>wsa:AttributedURI</u>
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="wsdl" type="wsa:AttributedURI" minOccurs="0"/>

element EntryType/type

diagram	
type	<u>admin:TypeType</u>
facets	enumeration dir enumeration entry
source	<xs:element name="type" type="admin:TypeType"/>

element EntryType/name

diagram	
type	<u>admin:EntryNameType</u>
source	<xs:element name="name" type="admin:EntryNameType"/>

simpleType ACLEntryType

namespace	urn:nces:messaging:1.2:types:admin
type	xs:string
used by	element <u>ACLTypr/member</u>
annotation	documentation Place holder for the allow/deny entries
source	<xs:simpleType name="ACLEntryType"> <xs:annotation> <xs:documentation>Place holder for the allow/deny entries</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"/> </xs:simpleType>

simpleType EntryNameType

namespace	urn:nces:messaging:1.2:types:admin
type	xs:string
used by	elements <u>entryName</u> <u>EntryType/name</u>
annotation	documentation The name of an entry
source	<xs:simpleType name="EntryNameType">



	<pre> <xs:annotation> <xs:documentation>The name of an entry</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"/> </xs:simpleType> </pre>
--	---

simpleType ScopeType

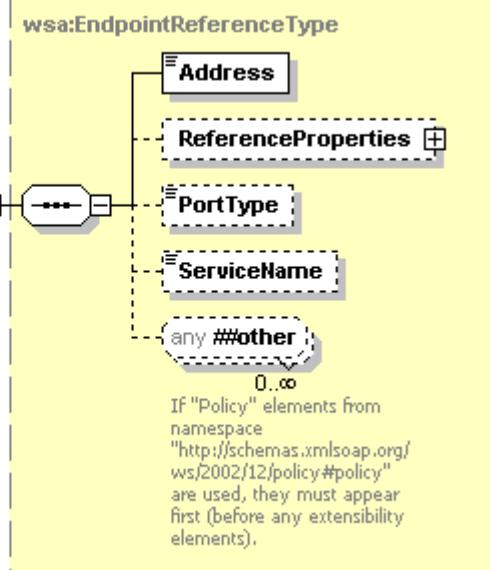
namespace	urn:nces:messaging:1.2:types:admin
type	restriction of xs:string
used by	elements scope ACLType/scoped
facets	enumeration subtree enumeration singlelevel enumeration object
annotation	documentation Scope for application of acl entry
source	<pre> <xs:simpleType name="ScopeType"> <xs:annotation> <xs:documentation>Scope for application of acl entry</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:enumeration value="subtree"/> <xs:enumeration value="singlelevel"/> <xs:enumeration value="object"/> </xs:restriction> </xs:simpleType> </pre>

simpleType TypeType

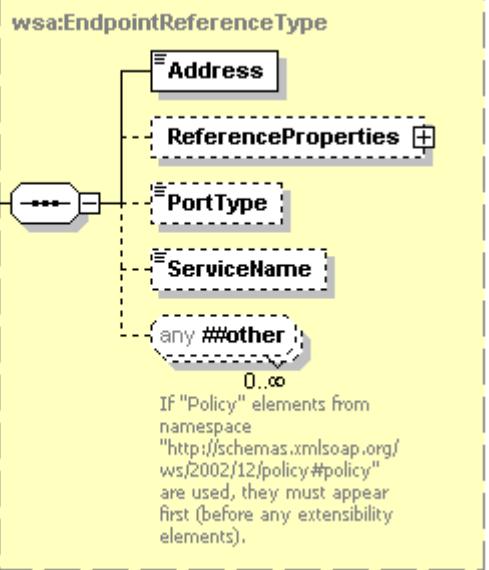
namespace	urn:nces:messaging:1.2:types:admin
type	restriction of xs:string
used by	element EntryType/type
facets	enumeration dir enumeration entry
source	<pre> <xs:simpleType name="TypeType"> <xs:restriction base="xs:string"> <xs:enumeration value="dir"/> <xs:enumeration value="entry"/> </xs:restriction> </xs:simpleType> </pre>

element wsa:Action

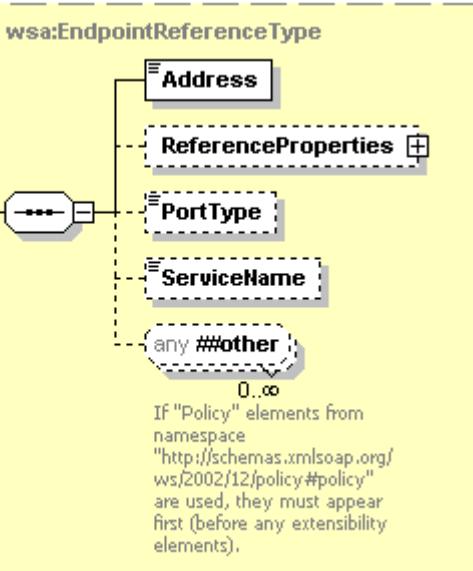
diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	wsa:AttributedURI
attributes	Name Type Use Default Fixed Annotation

source	<code><xss:element name="Action" type="wsa:AttributedURI"/></code>						
element wsa:EndpointReference							
diagram	 <pre> classDiagram class wsaEndpointReferenceType { Address ReferenceProperties PortType ServiceName "any ##other" } class wsaEndpointReference { --> wsaEndpointReferenceType } wsaEndpointReference --> wsaEndpointReferenceType </pre> <p>IF "Policy" elements from namespace "http://schemas.xmlsoap.org/ws/2002/12/policy#policy" are used, they must appear first (before any extensibility elements).</p>						
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing						
type	wsa:EndpointReferenceType						
children	Address ReferenceProperties PortType ServiceName						
attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> </table>	Name	Type	Use	Default	Fixed	Annotation
Name	Type	Use	Default	Fixed	Annotation		
source	<code><xss:element name="EndpointReference" type="wsa:EndpointReferenceType"/></code>						

element wsa:FaultTo

diagram							
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing						
type	wsa:EndpointReferenceType						
children	Address ReferenceProperties PortType ServiceName						
attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Name</td> <td>Type</td> <td>Use</td> <td>Default</td> <td>Fixed</td> <td>Annotation</td> </tr> </table>	Name	Type	Use	Default	Fixed	Annotation
Name	Type	Use	Default	Fixed	Annotation		
source	<xs:element name="FaultTo" type="wsa:EndpointReferenceType"/>						

element wsa:From

diagram	
---------	---

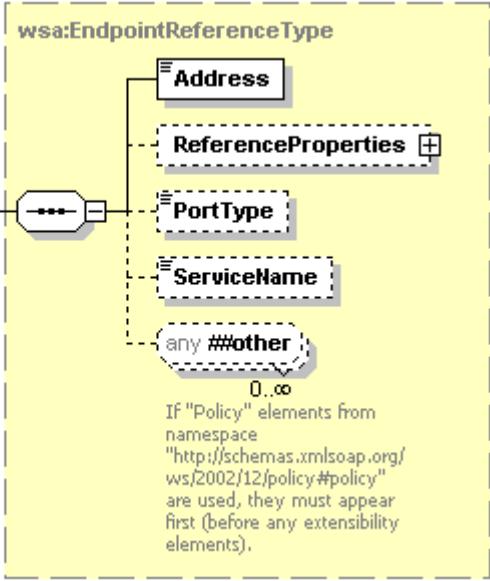


namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing					
type	<u>wsa:EndpointReferenceType</u>					
children	<u>Address</u> <u>ReferenceProperties</u> <u>PortType</u> <u>ServiceName</u>					
attributes	Name	Type	Use	Default	Fixed	Annotation
source	<xs:element name="From" type="wsa:EndpointReferenceType"/>					

element wsa:MessageID

diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	<u>wsa:AttributedURI</u>
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="MessageID" type="wsa:AttributedURI"/>

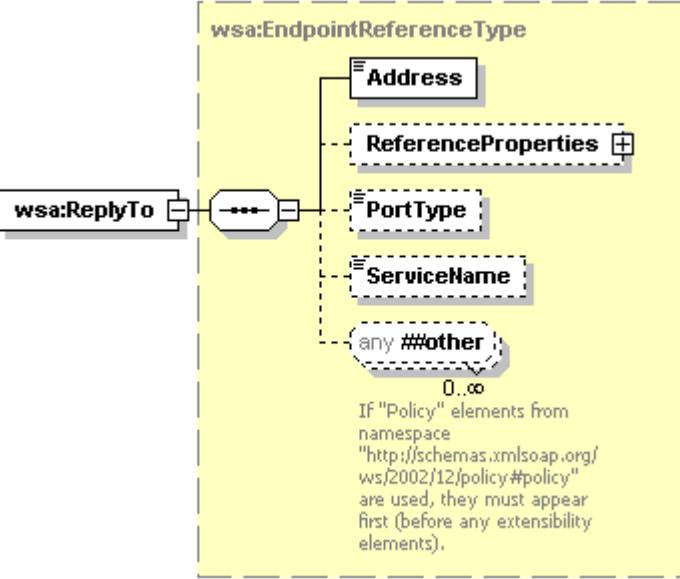
element wsa:Recipient

diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	<u>wsa:EndpointReferenceType</u>
children	<u>Address</u> <u>ReferenceProperties</u> <u>PortType</u> <u>ServiceName</u>
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="Recipient" type="wsa:EndpointReferenceType"/>

element **wsa:RelatesTo**

diagram													
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing												
type	<u>wsa:Relationship</u>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>RelationshipType</td> <td>xs:QName</td> <td>optional</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	RelationshipType	xs:QName	optional			
Name	Type	Use	Default	Fixed	Annotation								
RelationshipType	xs:QName	optional											
source	<xs:element name="RelatesTo" type="wsa:Relationship"/>												

element **wsa:ReplyTo**

diagram	 <p>wsa:EndpointReferenceType</p> <ul style="list-style-type: none"> Address ReferenceProperties PortType ServiceName any ##other <p>0..oo If "Policy" elements from namespace "http://schemas.xmlsoap.org/ws/2002/12/policy#policy" are used, they must appear first (before any extensibility elements).</p>						
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing						
type	<u>wsa:EndpointReferenceType</u>						
children	<u>Address</u> <u>ReferenceProperties</u> <u>PortType</u> <u>ServiceName</u>						
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> </table>	Name	Type	Use	Default	Fixed	Annotation
Name	Type	Use	Default	Fixed	Annotation		
source	<xs:element name="ReplyTo" type="wsa:EndpointReferenceType"/>						

element **wsa:To**

diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	<u>wsa:AttributedURI</u>



attributes	Name	Type	Use	Default	Fixed	Annotation
source	<xs:element name="To" type="wsa:AttributedURI"/>					

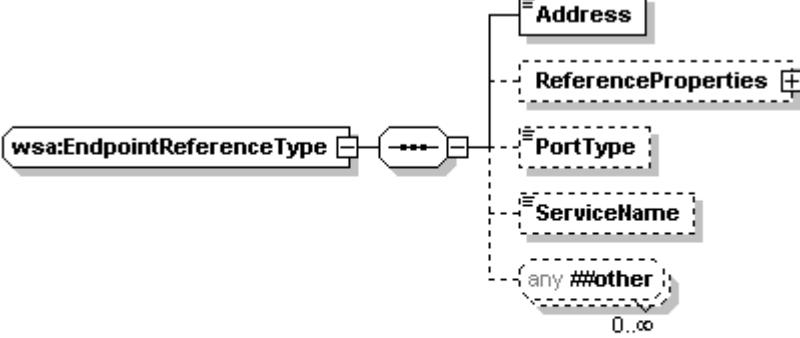
complexType wsa:AttributedQName

diagram	wsa:AttributedQName
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	extension of xs:QName
used by	element wsa:EndpointReferenceType/PortType
attributes	Name Type Use Default Fixed Annotation
source	<pre><xs:complexType name="AttributedQName"> <xs:simpleContent> <xs:extension base="xs:QName"> <xs:anyAttribute namespace="#other" processContents="lax"/> </xs:extension> </xs:simpleContent> </xs:complexType></pre>

complexType wsa:AttributedURI

diagram	wsa:AttributedURI
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	extension of xs:anyURI
used by	elements wsa:EndpointReferenceType/Address wsa:Action wsa:MessageID wsa:To EntryType/wsdl
attributes	Name Type Use Default Fixed Annotation
source	<pre><xs:complexType name="AttributedURI"> <xs:simpleContent> <xs:extension base="xs:anyURI"> <xs:anyAttribute namespace="#other" processContents="lax"/> </xs:extension> </xs:simpleContent> </xs:complexType></pre>

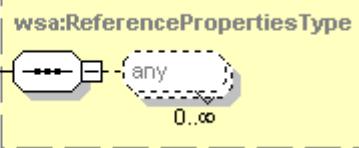
complexType wsa:EndpointReferenceType

diagram	 <p>If "Policy" elements from namespace "http://schemas.xmlsoap.org/ws/2002/12/policy#policy" are used, they must appear first (before any extensibility elements).</p>						
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing						
children	<u>Address</u> <u>ReferenceProperties</u> <u>PortType</u> <u>ServiceName</u>						
used by	elements <u>wsa:EndpointReference</u> <u>wsa:FaultTo</u> <u>wsa:From</u> <u>wsa:Recipient</u> <u>wsa:ReplyTo</u>						
attributes	<table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr> </thead> </table>	Name	Type	Use	Default	Fixed	Annotation
Name	Type	Use	Default	Fixed	Annotation		
source	<pre> <xs:complexType name="EndpointReferenceType"> <xs:sequence> <xs:element name="Address" type="wsa:AttributedURI"/> <xs:element name="ReferenceProperties" type="wsa:ReferencePropertiesType" minOccurs="0"/> <xs:element name="PortType" type="wsa:AttributedQName" minOccurs="0"/> <xs:element name="ServiceName" type="wsa:ServiceNameType" minOccurs="0"/> <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>If "Policy" elements from namespace "http://schemas.xmlsoap.org/ws/2002/12/policy#policy" are used, they must appear first (before any extensibility elements).</xs:documentation> </xs:annotation> <xs:any/> </xs:sequence> <xs:anyAttribute namespace="##other" processContents="lax"/> </xs:complexType></pre>						

element **wsa:EndpointReferenceType/Address**

diagram							
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing						
type	<u>wsa:AttributedURI</u>						
attributes	<table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr> </thead> </table>	Name	Type	Use	Default	Fixed	Annotation
Name	Type	Use	Default	Fixed	Annotation		

element **wsa:EndpointReferenceType/ReferenceProperties**

diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	<u>wsa:ReferencePropertiesType</u>
source	<xs:element name="ReferenceProperties" type="wsa:ReferencePropertiesType" minOccurs="0"/>

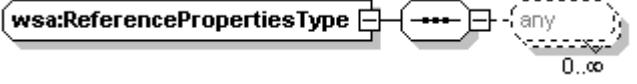
element **wsa:EndpointReferenceType/PortType**

diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	<u>wsa:AttributedQName</u>
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="PortType" type="wsa:AttributedQName" minOccurs="0"/>

element **wsa:EndpointReferenceType/ServiceName**

diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	<u>wsa:ServiceNameType</u>
attributes	Name Type Use Default Fixed Annotation PortName xs:NCName
source	<xs:element name="ServiceName" type="wsa:ServiceNameType" minOccurs="0"/>

complexType **wsa:ReferencePropertiesType**

diagram	
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
used by	element <u>wsa:EndpointReferenceType/ReferenceProperties</u>
source	<xs:complexType name="ReferencePropertiesType"> <xs:sequence> <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType>



complexType wsa:Relationship

diagram													
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing												
type	extension of xs:anyURI												
used by	element wsa:RelatesTo												
attributes	<table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr> </thead> <tbody> <tr> <td>RelationshipType</td><td>xs:QName</td><td>optional</td><td></td><td></td><td></td></tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	RelationshipType	xs:QName	optional			
Name	Type	Use	Default	Fixed	Annotation								
RelationshipType	xs:QName	optional											
source	<pre><xs:complexType name="Relationship"> <xs:simpleContent> <xs:extension base="xs:anyURI"> <xs:attribute name="RelationshipType" type="xs:QName" use="optional"/> <xs:anyAttribute namespace="##other" processContents="lax"/> </xs:extension> </xs:simpleContent> </xs:complexType></pre>												

complexType wsa:ServiceNameType

diagram													
namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing												
type	extension of xs:QName												
used by	element wsa:EndpointReferenceType/ServiceName												
attributes	<table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr> </thead> <tbody> <tr> <td>PortName</td><td>xs:NCName</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	PortName	xs:NCName				
Name	Type	Use	Default	Fixed	Annotation								
PortName	xs:NCName												
source	<pre><xs:complexType name="ServiceNameType"> <xs:simpleContent> <xs:extension base="xs:QName"> <xs:attribute name="PortName" type="xs:NCName"/> <xs:anyAttribute namespace="##other" processContents="lax"/> </xs:extension> </xs:simpleContent> </xs:complexType></pre>												

simpleType wsa:RelationshipTypeValues

namespace	http://schemas.xmlsoap.org/ws/2003/03/addressing
type	restriction of xs:QName
facets	enumeration wsa:Response
source	<pre><xs:simpleType name="RelationshipTypeValues"> <xs:restriction base="xs:QName"> <xs:enumeration value="wsa:Response"/> </xs:restriction></pre>



| `</xs:simpleType>`

XML Schema documentation generated with **XMLSPY** Schema Editor <http://www.altova.com/xmlspy>